DWeb Weekend 2025 - Saturday 8/16/2025

Roundtable Discussion 3.4

Title: *Commons infrastructure*
Time: *from - to*
Attendees: Dmitri, Santi, Eric, Kevin, Danny, Scott, Candida, Matt, Andreas, Louis
Note taker: Matthew Weidner

Please add your /initials/ at the front of a comment

General Notes

Participants' motivations: deploying commons; using commons; enabling apps that need a bit of trusted parties, storage, etc.

Q: "What are examples of commons services (for P2P)?"

- Point-to-point messaging
  - Response: Projects start out with desired for pure P2P, but later add bootstrap/routing server and makes things much easier
- Proxies & bridges. For when something needs a leader/access point to get to your community.
  - Ex: IPFS Gateway, TOR node
  - Content-based networking
- Store & forward
- Doc/file share, CDN
- Key recovery
- Public key directory
- Signaling servers (STUN)

- App store
  - Compose from public key directory + doc/file share
  - Trust in the app is the main challenge
- Notification service
  - Needed for smartphone apps (esp. iOS - mandates specific signaling server)
  - iOS processing time/memory limits can make this challenging (e.g. decryption/peering). Indexing nodes to help?
- Community search engines

What makes a service "commons"?

- Something trusted that we need to hold the protocol together
- Log-hanging fruit - service that is useful/necessary & inexpensive to operate; scarce resource is trust, not $$; ideally low risk
  - Anti-example: free email for nonprofits in the Bay Area. (B/c email already has cheap and long-lived providers.)
  - Look at lens of **necessity** - how much good does this gain over the centralized version? (E.g. Google Drive mostly okay)

Deep Dive: Public Key Directories

Easy engineering & cost wise; hard organizationally. So fits the low-hanging fruit criteria as well. (DZ: Especially good example.)

Example use case for key-based networking: Connect to camera in your garage by key, without knowing its IP address.

Trust in keys is critical - used for provenance and authentication.

- Verifying developers in app store
- Digital credentials from MIT
- Key directory is the map from public key to organization

DZ: Need separate architectures for public individuals & organizations vs private individuals. Latter can't be looked up.

Why EFF didn't do this: Single point of corruptibility. "No one should have this level of trust."

- Pick trusted entities powerful enough to handle the consequences vs use cases that are lighthearted enough to be okay with this (e.g. games).
- NGOs don't want to do it *for life and death cases*, but can be convinced for other cases (e.g. app signing)
- Concern about trusting startups to do KYC
    - Different KYC requirements for financial services vs other use cases. (Former costs $1-10 per verification.)
    - You're really trusting the government (these startups go back to govt DBs). Do we want to?
        - Gov't agencies don't want this responsibility. (Age verification DB attempt: Post Office → State Department → No; SSA → No, we want to get out of running the SSN DB.)

Summary to be reported to the other DWeb Weekend participants

a)

b)

c)

name of the reporter